

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Pengertian Sistem**

Sistem informasi berbasis komputer merupakan sekelompok perangkat keras dan perangkat lunak yang dirancang untuk mengubah data menjadi informasi yang bermanfaat. Jenis sistem informasi berbasis komputer, yaitu:

1. Pengolahan Data, Pengolahan data elektronik – *electronic data processing (EDP)* adalah pemanfaatan teknologi komputer untuk melakukan pengolahan data transaksi-transaksi dalam suatu organisasi. EDP adalah aplikasi sistem informasi akuntansi paling dasar dalam setiap organisasi. Sehubungan dengan perkembangan teknologi komputer, istilah pengolahan data mulai dikenal dan mempunyai arti yang sama dengan istilah EDP.
2. Sistem Informasi Manajemen (SIM), menguraikan penggunaan teknologi komputer untuk menyediakan informasi bagi pengambilan keputusan para manajer.
3. Sistem Pendukung Keputusan – *Decision Support Systems (DSS)*. DSS diarahkan untuk melayani permintaan informasi tertentu, khusus, dan tidak rutin dari manajemen. Contoh adalah penggunaan *spreadsheet* untuk melakukan analisis “*what if*” dari data operasi atau anggaran.

4. Sistem Pakar – *expert systems* (ES) adalah sistem informasi berbasis pengetahuan yang memanfaatkan pengetahuannya tentang bidang aplikasi tertentu untuk bertindak seperti seorang konsultan ahli bagi pemakainya.
5. Sistem Informasi Eksekutif – *executive information systems* (EIS). EIS dibuat bagi kebutuhan informasi stratejik manajemen tingkat puncak.
6. Sistem Informasi Akuntansi – sistem berbasis komputer yang dirancang untuk mengubah data akuntansi menjadi informasi (Agustinus; 2012 : 2-3).

## **II.2. Pengertian Informasi**

Informasi adalah data yang berguna yang telah diolah sehingga dapat dijadikan dasar untuk mengambil keputusan yang tepat. Informasi sangat penting bagi organisasi. Pada dasarnya informasi adalah penting seperti sumber daya yang lain, misalnya peralatan, bahan, tenaga, dsb. Informasi yang berkualitas dapat mendukung keunggulan kompetitif suatu organisasi.

Secara konseptual seluruh sistem organisasional mencapai tujuannya melalui proses alokasi sumber daya, yang diwujudkan melalui proses pengambilan keputusan manajerial. Informasi memiliki nilai ekonomis pada saat ia mendukung keputusan alokasi sumber daya, sehingga dengan demikian mendukung sistem untuk mencapai tujuan.

Pemakai informasi akuntansi dapat dibagi dalam dua kelompok besar: ekstern dan intern. Pemakai ekstern mencakup pemegang saham, investor, kreditor, pemerintah, pelanggan, pemasok, pesaing, serikat pekerja, dan masyarakat. Pemakai intern terutama para manajer, kebutuhannya bervariasi

tergantung pada tingkatannya dalam organisasi atau terhadap fungsi yang mereka jalankan (Agustinus ; 2012 : 1)

### **II.3. Pengertian Sistem Informasi Akuntansi**

Sistem informasi akuntansi adalah kumpulan sumber daya, seperti manusia dan peralatan, yang diatur untuk mengubah data menjadi informasi. Informasi ini dikomunikasikan kepada beragam pengambil keputusan. Sistem informasi akuntansi mewujudkan perubahan ini secara manual atau terkomputerisasi. SIA pada umumnya meliputi beberapa siklus pemrosesan transaksi:

1. Siklus pendapatan. Berkaitan dengan pendistribusian barang dan jasa ke entitas lain dan pengumpulan pembayaran-pembayaran yang berkaitan.
2. Siklus pengeluaran. Berkaitan dengan perolehan barang jasa dari entitas lain dan pelunasan kewajiban yang berkaitan.
3. Siklus produksi. Berkaitan dengan pengubahan sumber daya menjadi barang dan jasa.
4. Siklus keuangan. Kejadian-kejadian yang berkaitan dengan perolehan dan manajemen dana-dana modal, termasuk kas. (Agustinus ; 2012 : 3)

#### **II.3.1. Tujuan Sistem Informasi Akuntansi**

Lingkup sistem informasi akuntansi dapat dijelaskan dari manfaat yang didapat dari informasi akuntansi. Manfaat atau tujuan sistem informasi akuntansi tersebut adalah sebagai berikut :

1. Mengamankan harta / kekayaan perusahaan. Harta / kekayaan di sini meliputi kas perusahaan, persediaan barang dagangan, termasuk aset tetap perusahaan.
2. Menghasilkan beragam informasi untuk pengambilan keputusan. misal, pengelola toko swalayan memerlukan informasi mengenai barang apa saja yang diminati oleh konsumen. Membeli barang yang kurang laku berarti kas akan terjebak dalam persediaan dan berarti kehilangan kesempatan untuk membeli barang dagangan yang laku.
3. Menghasilkan informasi untuk pihak eksternal. Setiap pengelola usaha memiliki kewajiban untuk membayar pajak. Besarnya pajak yang dibayar tergantung pada omset penjualan (jika pengelola memilih menggunakan norma dalam perhitungan pajaknya) atau tergantung pada laba rugi usaha (jika pengelola memilih untuk tidak menggunakan norma dalam perhitungan pajaknya).
4. Menghasilkan informasi untuk penilaian kinerja karyawan atau divisi. Sistem informasi dapat juga dimanfaatkan untuk penilaian kinerja karyawan atau divisi.
5. Menyediakan data masa lalu untuk kepentingan audit (pemeriksaan). Data yang tersimpan dengan baik sangat memudahkan proses audit (pemeriksaan).
6. Menghasilkan informasi untuk penyusunan dan evaluasi anggaran perusahaan. Anggaran merupakan alat yang sering digunakan perusahaan untuk mengendalikan pengeluaran kas.

7. Menghasilkan informasi yang diperlukan dalam kegiatan perencanaan dan pengendalian. Selain berguna untuk membandingkan informasi yang berkaitan dengan anggaran dan biaya standar dengan kenyataan seperti yang telah dikemukakan (Anastasia Diana ; 2011 : 5-7).

#### **II.4. Pendapatan Jasa**

Pendapatan adalah arus masuk bruto dari manfaat ekonomi yang timbul akibat aktivitas normal perusahaan selama satu periode, arus masuk itu mengakibatkan kenaikan modal (ekuitas) dan tidak berasal dari kontribusi penanaman modal. Arus masuk dimaksud adalah hasil dari penjualan produk perusahaan.

Jasa adalah hasil penggunaan produk atau fasilitas perusahaan perusahaan berupa produk tidak berwujud. Untuk memproduksi jasa, perusahaan juga menggunakan bahan baku berupa tenaga kerja dan masukan modal. Biaya untuk karyawan, asuransi, kegiatan administrasi adalah contoh jasa antara lain penyewaan properti, mobil atau aset lain perusahaan.

Arus masuk bruto atau pendapatan adalah hasil dari penjualan produk yang hanya diterima dan dapat diterima oleh perusahaan. Jumlah yang ditagih atas nama pihak ketiga, seperti pajak pertambahan nilai, bukan merupakan manfaat ekonomi yang mengalir ke perusahaan dan tidak mengakibatkan kenaikan modal, dan karena itu harus dikeluarkan dari pendapatan.

Pengertian penjualan jasa bisa menyangkut pelaksanaan tugas yang secara kontraktual telah disepakati untuk dilakukan selama suatu periode waktu. Jasa tersebut dapat diserahkan selama satu periode atau lebih. Penghasilan bersih dari

penjualan jasa adalah penghasilan setelah dikurangi semua komisi penjualan, retur, rabat, diskon dan sebagainya (Kuswadi ; 2010 : 58-59).

## II.5. Bentuk Single Step

Laporan laba/rugi adalah laporan yang menunjukkan besarnya pendapatan dan beban pada akhir periode akuntansi. Penyusunan laporan laba/rugi yang bersumber dari kertas kerja, datanya diambil dari kolom perhitungan laba/rugi. Penyusunan laporan laba/rugi dapat dibedakan atas dua bentuk, yaitu bentuk langsung (*single step*) dan bentuk bertahap (*multiple step*).

Laporan laba rugi *single step*, adalah laporan laba rugi yang penyajiannya tidak membedakan antara pendapatan usaha utama dengan pendapatan diluar usaha utama (Bina Sarana Informatika : 1)

Jadi: Pendapatan Jasa – Beban = Laba/Rugi

Ibnu Delly TV Service Laporan Rugi Laba Per 31 Desember 200X		
Pendapatan Jasa		Rp. XXX
Biaya Usaha:		
Biaya perlengkapan	Rp. XXX	
Biaya gaji dan upah	Rp. XXX	
Biaya serba-serbi	Rp. XXX	
	-----	
Total Biaya Usaha		Rp. XXX -----
Laba bersih		Rp. XXX ----- -----

**Gambar. II.1 Contoh Laporan Laba/Rugi Bentuk Single Step  
(Sumber : Bina Sarana Informatika)**

## II.6. Visual Basic

*Visual Basic* dibuat oleh *Microsoft*, merupakan salah satu bahasa pemrograman berorientasi objek yang mudah dipelajari. Selain menawarkan kemudahan, *Visual Basic* juga cukup andal untuk digunakan dalam pembuatan berbagai aplikasi, terutama aplikasi *database*. *Visual basic* merupakan bahasa pemrograman *event drive*, di mana program aplikasi yang dapat berupa kejadian atau *event*, misalnya ketika *user* mengklik tombol atau menekan enter. Jika kita membuat aplikasi dengan *Visual Basic* maka kita akan mendapatkan *file* yang menyusun aplikasi tersebut, yaitu :

1. *File Project (\*.vbp)*

*File* ini merupakan kumpulan dari aplikasi yang kita buat. *File project* bisa berupa *file* \*.frm, \*.dsr atau file lainnya.

2. *File Form (\*.frm)*

*File* ini merupakan *file* yang berfungsi untuk menyimpan informasi tentang bentuk *form* maupun *interface* yang kita buat (Edy Winarno ; 2010 : 83).

## II.7. Database

*Database* adalah suatu kumpulan data yang terintegrasi yang diorganisasikan untuk memenuhi kebutuhan para pemakai didalam suatu organisasi. semua pengaksesan ke *database* ditangani oleh suatu piranti lunak yang disebut *database management system*. Data tersebut diorganisasikan sedemikian rupa agar tidak terjadi duplikasi yang tidak perlu, sehingga dapat diolah atau dieksplorasi secara cepat dan mudah untuk menghasilkan informasi.

Secara fisik suatu *database* terdiri atas *entity* yang biasanya disebut *table*. Tiap-tiap *table* tersebut mempunyai atribut-atribut yang disebut *field*. Dan isi dari *table* tersebut disebut *record* yang merupakan tumpukan baris yang mempunyai minimal satu atribut dari *table* tersebut.

*Database* rasional adalah salah satu *database* yang sering digunakan dan tersedia untuk aplikasi komputer yang penting. *Database* relasional menyediakan sebuah kemampuan yang sangat kuat dan fleksibel untuk menyimpan data dari berbagai jenis (Luther ; 2011 : 132).

## **II.8. SQL Server 2008**

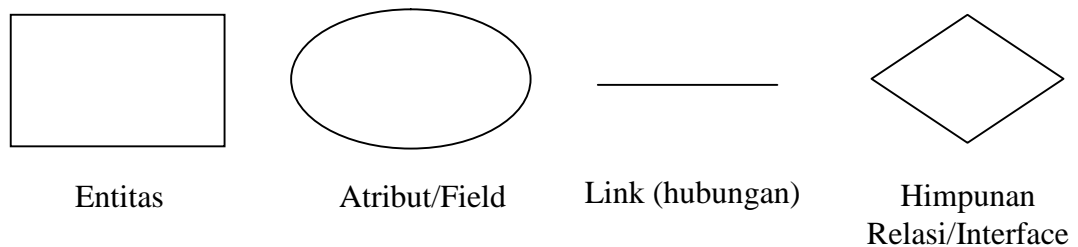
*SQL Server* 2008 adalah sebuah terobosan baru dari Microsoft dalam bidang *database*. *Sql server* adalah DBMS (*Database Management System*) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan Oracle. *Sql server* 2008 dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa *Sql server* 2008 membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data.

Microsoft merilis *Sql server* 2008 dalam beberapa versi yang disesuaikan dengan segment-segment pasar yang dituju. Versi-versi tersebut adalah sebagai berikut. Menurut cara pemrosesan data pada prosesor maka Microsoft mengelompokkan produk ini berdasarkan 2 jenis yaitu :

- a. *Versi 32-bit(x86)*, yang biasanya digunakan untuk komputer dengan single prosesor (Pentium 4) atau lebih tepatnya prosesor 32 bit dan sistem operasi Windows XP.
- b. *Versi 64-bit(x64)*, yang biasanya digunakan untuk komputer dengan lebih dari satu prosesor (Misalnya *Core 2 Duo*) dan system operasi 64 bit seperti Windows XP 64, Vista, dan Windows 7 (Wenny Widya ; 2012 : 3)

## II.9. *Entity Relationship Diagram (ERD)*

*Entity Relationship Diagram* atau ERD adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antarentitas. Proses memungkinkan analisis menghasilkan struktur basisdata yang baik sehingga data dapat disimpan dan diambil secara efisien (Janner Simarmata ; 2010 : 67).



**Gambar. II.2 Bentuk Simbol ERD**  
(Sumber : Janner Simarmata ; 2010 : 67)

## II.10. Teknik Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan

menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional.

### II.10.1. Bentuk-bentuk Normalisasi

a. Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti *format* tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaanya.

b. Bentuk normal tahap pertama (1<sup>st</sup> Normal *Form*)

Definisi :

Sebuah table disebut 1NF jika :

- Tidak ada baris yang duplikat dalam tabel tersebut.
- Masing-masing cell bernilai tunggal

Catatan: Permintaan yang menyatakan tidak ada baris yang duplikat dalam sebuah tabel berarti tabel tersebut memiliki sebuah kunci, meskipun kunci tersebut dibuat dari kombinasi lebih dari satu kolom atau bahkan kunci tersebut merupakan kombinasi dari semua kolom.

c. Bentuk normal tahap kedua (2<sup>nd</sup> normal *form*)

Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam *primary key* memiliki ketergantungan fungsional pada *primary key* secara utuh.

d. Bentuk normal tahap ketiga (3<sup>rd</sup> normal *form*)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi  $X \rightarrow A$ , dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka :

- X haruslah *superkey* pada tabel tersebut.
- Atau A merupakan bagian dari *primary key* pada tabel tersebut.

e. Bentuk Normal Tahap Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk Normal keempat berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form* (PJNF).

f. Boyce Code Normal *Form* (BCNF)

- Memenuhi 1<sup>st</sup> NF
- Relasi harus bergantung fungsi pada atribut *superkey* (Janner Simarmata ; 2010 : 76).

## II.11. UML (*Unified Modeling Language*)

Menurut Windu Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.


UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

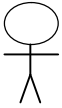

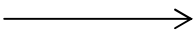

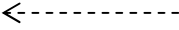
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

### 1. *Use case* Diagram

*Use case* diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

**Tabel II.1. Simbol *Use Case***

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>




	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki control terhadap <i>use case</i>.</p>
	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.</p>
	<p><i>Include</i>, merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.</p>
	<p><i>Extend</i>, merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.</p>

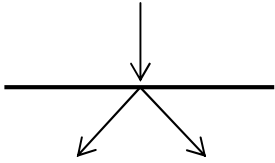
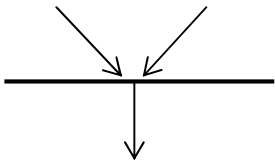
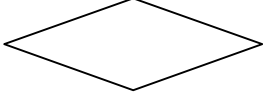
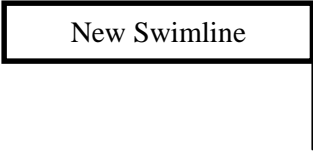
(Sumber : Windu Gata ; 2013 : 4)

## 2. Diagram Aktivitas (*Activity Diagram*)

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

**Tabel II.2. Simbol *Activity Diagram***

Gambar	Keterangan
	<p><i>Start point</i>, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.</p>
	<p><i>End point</i>, akhir aktifitas.</p>
	<p><i>Activites</i>, menggambarkan suatu proses/kegiatan bisnis.</p>

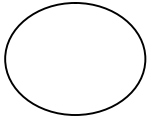
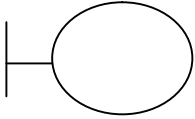
	<p><i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.</p>
	<p><i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.</p>
	<p><i>Decision Points</i>, menggambarkan pilihan untuk pengambilan keputusan, <i>true</i>, <i>false</i>.</p>
	<p><i>Swimlane</i>, pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.</p>

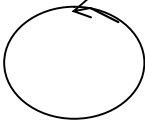

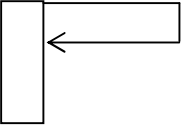
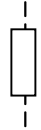

(Sumber : Windu Gata ; 2013 : 6)

### 3. Diagram Urutan (*Sequence Diagram*)

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

**Tabel II.3. Simbol *Sequence Diagram***

Gambar	Keterangan
	<p><i>EntityClass</i>, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.</p>
	<p><i>Boundary Class</i>, berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.</p>

	<p><i>Control class</i>, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.</p>
	<p><i>Message</i>, simbol mengirim pesan antar <i>class</i>.</p>
	<p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>
	<p><i>Activation</i>, <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

(Sumber : Windu Gata ; 2013 : 7)

#### 4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

*Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

**Tabel II.4. Multiplicity Class Diagram**

<b>Multiplicity</b>	<b>Penjelasan</b>
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

*(Sumber : Windu Gata ; 2013 : 9)*